

Piece
- piece: String - pos: pair<int,int> - isKing: bool
+ Piece(String, int, int) + getPos(): pair<int,int> + setPos(int, int): void + getPiece(): String + setPiece(String): void + setKing(bool): void + getKingStatus(): bool + swapPos(pair<int,int>): void

Node
+ data: int
+ left: Node *
+ right: Node *

Player
- name: String - ourPiece: String - board: Board *
+ Player(String) + setBoard(Board *): void + setPiece(String): void + getPiece(): String + getName(): String + makeMove(map<string,int> &, Tree *): void

Board
- DIM: const int - board: list<list<Piece *>>
- fillBrd(): void - isVldSquare(int, int): bool - putPieces(): void - canMove(String, pair<int,int>): bool - canMoveKing(pair<int,int>): bool - canCapture(String, pair<int,int>, pair<int,int> &, bool &): bool - canCaptWithKng(String, pair<int,int>, pair<int,int> &, bool &): bool - capture(String, pair<int,int>, pair<int,int>, pair<int,int>): void - dsplyKings(String): void - swap(pair<int,int> &, pair<int,int> &): void - partition(vector<pair<int,int>> &, int, int): int - quickSort(vector<pair<int,int>> &, int, int): void - willForfeit(String): void + Board() + ~Board() + dsplyBrd(): void + jumpTo(int, int): list<Piece *>::iterator + move(String, map<String,int> &, Tree *)

Tree
- root: Node *
- newNode(int): Node * - prntPre(Node *): void - prntIn(Node *): void - prntPost(Node *): void - insert(Node *&,int): void - destroy(Node *&): void - height(Node *): int - delNode(Node *&,int): void - findMin(Node *): Node * - lftRot(Node *&): void - rghtRot(Node *&): void - lftRghtRot(Node *&): void - rghtLftRot(Node *&): void - balance(Node *&): void + Tree() + Tree(int) + ~Tree() + prntPre(): void + prntIn(): void + prntPost(): void + prntLvl(): void + empty(): bool + insert(int): void + fill(int): void + height(): int + delNode(int): void + findMin(): Node *